Secure Playback using OP-TEE

Overview
Problem Statement

- Want to open browser, login and play secure DRM videos
- No one in middle or even after rooting target device can access decrypted/decoded content.
- Performance / Zero-copy
Session Objectives

- Basic building blocks for Secure Playback, their interaction
- Pointers to related sessions at HKG15
Hardware that makes this possible

- ARM® TrustZone® Technology
- TrustZone Address Space Controller (TZASC)
- TrustZone Protection Controller (TZPC)
Hardware: ARM® TrustZone®

Hardware: TrustZone Address Space Controller (TZASC)
Hardware: TrustZone Protection Controller (TZPC)

Reference: ARM Security Technology - Building a Secure System using TrustZone Technology
ARM® TrustZone® and OPTEE

Reference: http://www.slideshare.net/linaroorg/lcu14-500-arm-trusted-firmware
What must happen
What must Happen
DRM System

- A DRM Solution provides capability to license, securely distribute protect content.
- Eg: Google® Widevine® DRM, Microsoft® PlayReady® DRM
- CDM session APIs wrap a DRM solution.
W3C EME

- Extends HTMLMediaElement providing APIs to control playback of protected content.
- Provides API that enables Web Apps to interact with DRM Provider and CDM
- Browser needs to implement EME extensions.
- EME does not enforce which DRM should be used.
Reference: http://www.w3.org/TR/encrypted-media/
W3C EME: How does it fit?
V4L2 based HW accelerated Decoder

- V4L2 provides interface at kernel boundary for Video Codecs (mem-to-mem devices)
  - Typical MPEG4 decode use case would involve just few system calls.
- Queueing/Dequeueuing buffer to OUTPUT and CAPTURE queues using ioctls once the initial formats and buffers are setup (again using open/ioctl/mmap)
- All State Management (play/pause) expected to be handled higher up in user space. Eg: GStreamer / StageFight. Not at V4L2 layer and below.
- Supports buffer queue/dequeue of dmabuf FDs.
- V4L2 Client Driver talks to the HW accelerators.
- Ref: Video4Linux2: Path to a Standardized Video Codec API
GStreamer

● Pipeline based Media framework ([link](http://gstreamer.freedesktop.org/))
● Allows to create variety of media-handling software components like
  ○ Simple audio/video playback/record
  ○ Streaming audio/video playback/record
  ○ Audio/Video Editing
● GStreamer provides libraries and plugins for
  ○ Various codecs
    ▪ Pure Software: Eg: FFmpeg, vorbis
    ▪ Hardware Accelerated Codecs: Eg: *Via V4L2 or equivalent
  ○ Audio/Video Synchronization
  ○ Mux / Demux etc
● Ref: http://gstreamer.freedesktop.org/
Chromium - Chrome - PPAPI

- **Chromium** is the open-source web browser project from which Google Chrome draws its source code.
- **Native Client** is a sandboxing technology for safe execution of platform-independent untrusted native code in a web browser.
- **Pepper** started at Google as a way to address portability and performance issues with NPAPI, particularly for out of process plugins.
GStreamer/V4L2: How it fits
And it all comes together :)
The Map
The Map

HKG15-106
Related Presentations (TBD: Add links)

- HKG15-106: Replacing CMEM: Gil Pitney (2/9 3PM)
- EME Implementation in Chromium: Linaro Clear Key: Matt Snoby (2/12 11:15AM)
- HKG15-403: Chromium Blink on Wayland with HW accelerated video playback: Zoltan, Christophe (2/12 10:10AM)
- HKG15-304 Secure Media using DMA-buf: Benjamin Gainard (**Cancelled**)
References

- ARM® TrustZone®
- ARM® Security Technology - Building a Secure System using TrustZone Technology
- ARM Trusted Firmware LCA14
- http://www.w3.org/TR/encrypted-media/
- http://gstreamer.freedesktop.org/
- Video4Linux2: Path to a Standardized Video Codec API
- Chromium
- Chromium Native Client (NaCl), PPAPI
Backup Slides